

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Martin Põhjakivi

**Pärmi genoomi funktsioonide
visualiseerimine veebirakenduses**

Bakalaureusetöö (9 EAP)

Juhendajad: Rahul Kumar
Sven Aller

Tartu 2019

Pärmi genoomi funktsioonide visualiseerimine veebirakenduses

Lühikokkuvõte:

Käesoleva bakalaureusetöö eesmärgiks on geneetikast huvitatud inimestele luua tööriist veebirakenduse näol, mis aitab genoomis väljenduvate funktsioonide asukohta määrata. Töös antakse ülevaade varasema info hankimise viiside ning andmeallikate kohta. Lisaks käsitletakse valitud tehnoloogiaid, mis kasutusele võeti ning ka valminud tööriista.

Võtmesõnad:

Genoom, annotatsioonid, visualiseerimine, veebirakendus

CERCS: P175, Informaatika

Abstract:

The goal of the present Thesis is to create a tool for people interested in genetics, to help determine the location of a given function of the genome on the genome's chromosomes. The Thesis gives an overview of the earlier method of getting such information and of the relevant data sources. The Thesis also goes over the choices of technology and the resulting tool itself.

Keywords:

Genome, annotations, visualization, web application

CERCS: P175, Informatics

Sisukord

| | | |
|-----|---|----|
| 1. | Sissejuhatus | 5 |
| 2. | Mõisted ja terminid | 6 |
| 3. | Geeniontoloogia ja pärm | 7 |
| 3.1 | Geeniontoloogia konsortsiumi geeniontoloogia projekt..... | 7 |
| 3.2 | Pärmi genoomi andmebaas | 8 |
| 3.3 | Vana protsessi kasutatavus | 8 |
| 4. | Nõuded | 10 |
| 4.1 | Funktsionaalsed nõuded..... | 10 |
| 4.2 | Mittefunktsionaalsed nõuded | 11 |
| 5. | Tehniline lahendus | 12 |
| 5.1 | Eesrakendus | 12 |
| 5.2 | Tagarakendus | 15 |
| 5.3 | Andmebaas..... | 16 |
| 6. | Andmete ettevalmistamine | 17 |
| 6.1 | Geenide asukohad kromosoomidel | 18 |
| 6.2 | Geenide kuuluvus geeniontoloogia termini alla..... | 18 |
| 6.3 | Geeniontoloogiate nimed..... | 19 |
| 7. | Rakenduse pakendamine, serveerimine ning majutus | 20 |
| 7.1 | Veebi ning rakenduse serverid | 20 |
| 7.2 | Majutus | 22 |
| 7.3 | Kasutajaliidese kirjeldus | 24 |
| 8. | Testimine..... | 26 |
| 8.1 | Veebibrauserite tugi..... | 26 |
| 8.2 | Jõudlus | 26 |
| 8.3 | Tagasiside kasutajatelt | 29 |

| | | |
|-----|-------------------------|----|
| 9. | Kokkuvõte | 30 |
| 10. | Viidatud kirjandus..... | 31 |
| | Lisad | 33 |
| I. | Litsents | 33 |

1. Sissejuhatus

Geenitehnoloogias ja muudes bioloogia ning keemia uurimisvaldkondades tegeletakse paljudel erinevatel abstraktsustasemetel uuritavate objektide omaduste ning funktsioonide kirjeldamisega. Uurimusi läbiviies võib aga kohati ette tulla situatsioone, kus uurimiseks on vaja infot omavahel seotud olemasolevatest allikatest, mille vahel puudub kiire viis vajalike seoste kättesaamiseks.

Geneetika ning muud geenidega kokkupuutuvad distsipliinid pakuvad teadlastele huvi, kuna nende abil on meil võimalik erinevate organismide funktsioneerimist jagada väiksemateks funktsionaalseteks osadeks, mis aitab tuua selgust paljude nähtuste põhjuslikkusele.

Arengud genoomide sekveneerimises ning transkriptoomikas on avanud teadlaste jaoks palju uusi võimalusi ning sellega on kaasnenud huvi kasv erinevate funktsionaalsuste eest vastutavate üksuste isoleerimise vastu, mis aitab kaasa näiteks efektiivsemale geeni-siirdamisele.

Kuna imetajate genoomi baasfunktsionaalsused on väga sarnased pärmi omadega ning pärmi funktsioone on läbi aastakümnete detailselt uuritud, on pärm hea mudelorganism, mille peal teha erinevaid uuringuid. Seetõttu piirdub ka antud töö pärmi genoomiga.

Antud töö eesmärgiks on implementeerida tööriist, mis aitab saada võrreldes olemasolevate allikatega kiiremini ülevaadet mõne geeniontoloogia termini alla kuuluvate geenide asukohtadest pärmi genoomis. Töö raames valmis andmebaas koos kasutajale suunatud veebisaidiga, kus saab teha vastavaid päringuid.

Käesoleva töö kirjalikus osas antakse esmalt ülevaade tööriistaga seotud bioloogia taustast. Seejärel käsitletakse lahendatavat probleemi vana protsessi kontekstis olemasolevate tööriistadega. Edasi esitatakse uue lahenduse nõuded ning implementeerimiseks vajaminevate tarkvarade valikuvõimalused koos valikutega. Kirjaliku osa lõpetab valminud rakenduse majutamise seotud otsuste kirjeldus ning valminud lahenduse nõuetele vastavuse testi tulemuste esitus.

2. Mõisted ja terminid

Geen - Geenides asetseb organismide pärivusega edasikanduv info. Eraldiseisvana paikneb geenides olev DNA biheeliks struktuuris, mis tähendab, et kaks DNA ahelat, mis koosnevad „A“, „T“, „C“, „G“ ühikutest, on omavahel paardunud „A“ ühik „T“ ühikuga ning „G“ ühik „C“ ühikuga.

Aluspaar - Igat geenis asetsevat kahe DNA ahela ühenduskohta, näiteks „A“ ühik „C“ ühikuga, nimetatakse üheks aluspaariks. Seega uurides DNA biheeliks struktuure, võib öelda, et üks konkreetne geen koosneb näiteks 5 000 aluspaarist.

Genoom – Kogu ühe organismi unikaalsete geenide kogumit nimetatakse genoomiks. Organismides nagu pärm või imetajad, koosneb genoom fikseeritud arvust kromosoomidest. Kromosoom omakorda on eraldiseisev üksus, kus on üks väga pikk DNA biheeliks struktuur, kuhu on kodeeritud tuhandeid geene. Seega rääkides kromosoomide suurustest, võib näiteks väita, et mõni konkreetne kromosoom koosneb 2 000 000 aluspaarist ning et mõne konkreetse geeni asukoht antud kromosoomil on aluspaaridel 100 000 kuni 102 500.

Päristuumsed – ehk eukarüootideks nimetatakse organismide rühma, millel on selgelt defineeritud rakutuum ning kus asub ka kromosoomidena geneetiline informatsioon [1].

3. Geeniontoloogia ja pärm

Nagu paljudes distsipliinides, on ka molekulaarbioloogia kirjeldavas harus standarditel ning ühtsetel arusaamadel suur roll. Eriti ilmsiks on see fakt tulnud geneetikas päristuumsete kirjeldamisel, kus on olemas juba mitmeid terviklike mudeleid erinevate organismide genoomide kohta.

Esimeste päristuumsete genoomide võrdlusest on tulnud välja, et väga suur osa vastavate organismide geenide poolt kodeeritavatest valkudest täidavad samasid või sarnaseid funktsioone. Paljud kattuvad funktsioonid kuuluvad nõ bioloogiliste protsesside tuumikusse.

Viimastel aastakümnetel on palju eksperimentaalseid tõestusi leidnud asjaolu, et seni uuritud mitte-imetajatest mudelorganismid jagavad baasfunktsionaalsusi paljuti ka imetajatega. Eriti on taoliste järeldusteni jõutud ainurakse pagaripärmi (*Saccharomyces cerevisiae*, edaspidi pärm) puhul [2].

3.1 Geeniontoloogia konsortsiumi geeniontoloogia projekt

Arvestades genoomide, geenide ja nende poolt väljenduvate funktsionaalsuste poolt loodud süsteemide keerukust, on Geeniontoloogia konsortsium (*GOC*) tegelenud kahe mugavalt töödeldava ressursi loomisega [3]. Nendeks on:

- Geeniontoloogia (*GO*) – kirjeldab bioloogiliste funktsioonide loogilist atsüklilist struktuuri
- *GO* annotatsioonid – tõendatud väited *GO* termide (*GOT*) ning geenide vahel [4]

Antud ressursi juurde kuulub ka informatsioon pärmi kohta, mille genoomi [5] esiletõusvateks omadusteks on:

- Koosneb 16 kromosoomist
- Sisaldab 12 156 677 aluspaari
- Sisaldab 6275 geeni

3.2 Pärmigenoomi andmebaas

Pärmigenoomi andmebaas (The *Saccharomyces* Genome Database; *SGD*) [6] kujutab endast läbi veebi kättesaadavat kogumit erinevatest allikatest ning tööriistadest. Keskkond sai alguse Stanfordini ülikoolist aastal 1993 eesmärgiga pärmiga töötavatel teadlastel lihtsustada andmete kättesaamist mahukatest allikatest.

Keskkonnas leitavad genoomiga tegelevad tööriistad kasutavad GO poolt kirjeldatud struktuure [7]. Järgnevalt toome välja mõned peamised tööriistad, mida antud keskkonnas on võimalik kasutada:

1. *YeastMine* – otsingumootor, mis aitab leida ning analüüsida infot geenide, valkude omaduste, GO annotatsioonide, kromosomaalsete funktsioonide, seotud kirjanduse ning teiste andmetüüpide kohta [8].
2. *GBrowse* – keskkond, kus saab visualiseerida igat pärmikromosoomi, nähes seal paiknevaid gene [9].
3. *GO Term Finder* – otsingumootor, kus saab geenide järgi leida milliste *GOT* alla nad kuuluvad [10].

Eelnimetatud ning teised tööriistad, mida *SGD* keskkond pakub tulevad kasuks paljudes situatsioonides, kuid nad ei lahenda ükski efektiivselt *GOT* alla kuuluvate geenide asukohtade leidmise ülesannet.

3.3 Vana protsessi kasutatavus

Siinkohal esitatakse lühidalt käesoleva projekti poolt lahendatav kasutatavuse probleem olemasolevate tööriistade juures. Esmalt tuuakse välja protsessi läbiviimiseks vajalikud sisendid ning väljundid, seejärel pakutakse näidissisend, misjärel esitatakse tegevusetappe kirjeldav diagramm.

Vajalikud sisendid:

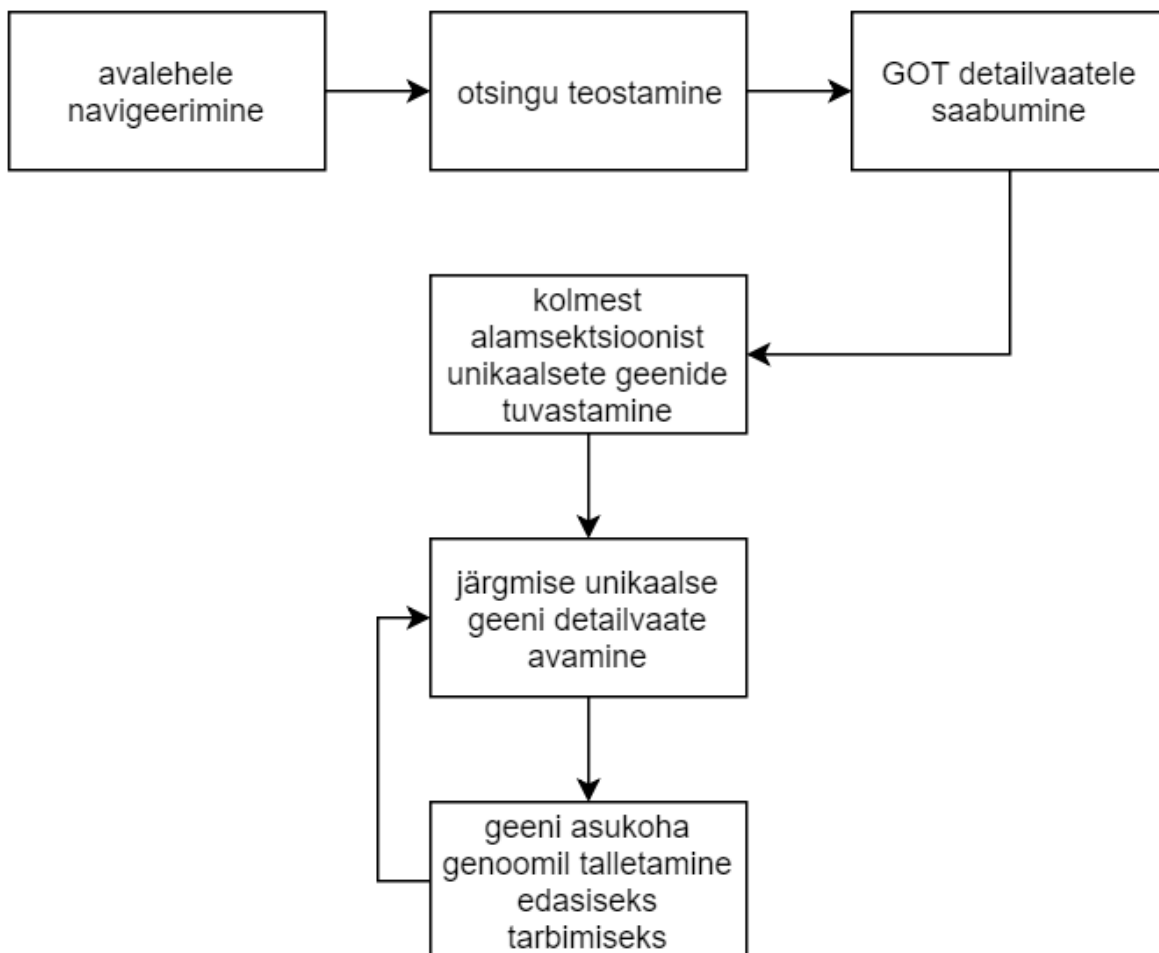
- *GOT* nimi
- *GOT* kohta käiv identifitseerimistunnus „*GO ID*“

Oodatav väljund on nimekiri geenidest koos järgnevate tunnustega iga geeni kohta:

- Geeni nimi
- Geeni SGD identifikaator
- Kromosoomi number
- Paiknemisvahemik kromosoomi aluspaaridel

Protsessi sisendiks sobib näiteks identifitseerimistunnus „GO:0005576“ („*extracellular region*“).

Joonisel 1 on esitatud etapid, mis tuleb SGD keskkonnas läbi teha, et vajalik info kätte saada. Esitatud protsessist võib näha, et väljatoodud tegevus ei ole optimaalne, arvestades andmeid, millele olemasolevatel tööriistadel ligipääs on. Siinkohal tasub märkida, et ühe ontoloogia-termini alla kuuluvate nime järgi unikaalsete geenide arv võib olla üle saja, olenevalt terminist. Lisaks puudub võimalus protsessi lõpus saada ülevaadet võimalikest asukohtade klastritest, kui just vastavat analüüsi ise eraldi läbi ei vii.



Joonis 1. Vana protsessi diagramm

4. Nõuded

Tööriista veebiliidese nõuded sai kokku lepitud juhendajaga, kes kohati kaasas arutellu ka kolmandaid osapooli oma valdkonnast.

4.1 Funktsionaalsed nõuded

Üldine

1. Esilehel peab olema lühitutvustus tööriista kohta.
2. Esilehel peab olema juhend, tööriista kasutamise kohta.

Otsing

Otsingulahtri eesmärgiks antud rakenduses on vastu võtta sisendina geeniontoloogia identifikaatoreid kahes formaadis (sgd keskkonnas kasutatava „GO:“ prefiksiga ning puhtnumbrilisel kujul) ning otsingute käivitamine ilma eraldi nupuvajutust vajamata.

1. Kasutaja peab saama sisestada otsingusse vabateksti.
2. Otsing peab koosnema ühest vabatekstiväljast.
3. Otsing peab tühjalt näitama näidet korrektse sisendiga.
4. Otsing peab korrektse sisendina vastu võtma numbritest koosnevat teksti.
5. Otsing peab korrektseks lugema sisendi, mis on prefiksiga „GO:“.
6. Otsing peab vastu võtma maksimaalselt 50 tähemärki.
7. Otsing peab kõik teised sisendid lugema ebakorrektses.
8. Kasutaja peab nägema teavitust ebakorrektses sisendi kohta.
9. Otsing peab automaatselt käivituma sisendi muutumisel.
10. Otsing ei tohi käivituda tihedamini kui üks kord poole sekundi jooksul.

Väljundi graafi osa

Graafi eesmärgiks on anda esmane ülevaade otsingutulemustest. Huvipakkuvateks iseärasusteks võib pidada jaotusi nii üle mitme kromosoomi kui ka ühe kromosoomi siseselt.

1. Graafi vertikaal-skaala peab kirjeldama kromosoomide nimesid.
2. Graafi horisontaal-skaala peab kirjeldama aluspaaride arvu.
3. Graafi peal peab olema näha staatiliselt ühe kihina iga kromosoomi täispikkus.
4. Graafi peal peab olema näha otsingutulemuses olevad asukohad korrektse sisendi puhul.

5. Ebakorrekse või tühja sisendi puhul peab graafi asukohtade kiht olema tühi.

Väljundi tabeli osa

Kuna tulemuste arv võib olla suur, tuleb kasuks andmete esitamine kergesti teisaldataval kujul.

- Tabeli päised peavad kirjeldama järgmiseid veergusid: geen, kromosoom, alguspunkt, lõpp-punkt.
- Tabeli read peavad olema täidetud ainult korrekse sisendi puhul.

4.2 Mittefunktsionaalsed nõuded

Mõõdetavad nõuded

- Veebiliides peab esialgse seisundi laadima maksimaalselt 5 sekundiga.
- Otsing peab andma tulemuse maksimaalselt poole sekundiga, mitte arvestades otsingulahtri enda poolesekundilist reageerimispuhvrit.
- Veebiliides peab ühilduma enimkasutatud veebibrauseritega
- Veebiliides peab ühilduma enimkasutatud veebibrauseritega mobiilsetel platvormidel

Hinnangulised nõuded

- Veebiliidese funktsionaalsus peab olema intuitiivne

5. Tehniline lahendus

Järgnevalt antakse ülevaade tehnoloogiate valikutest. Käsitletakse eesrakendust, visualiseerimise teeki, tagarakendust ning andmebaasi. Iga osa juures tuuakse välja ka alternatiive ning põhjendatakse lühidalt valikut.

Veebileht on töö kirjutamise hetkel kättesaadav aadressil <http://pohjakivi.ee/>.

5.1 Eesrakendus

Eesrakenduseks (*front-end application*) nimetatakse rakenduse osa, millega kasutaja otseselt kokku puutub. Nimelt kuuluvad siia alla läbi presentatsioonikihi esitatavad kuvad, graafika ning kasutaja sisendi töötlemine. Veebis domineerivad eesrakenduse puhul tehnoloogiad HTML, CSS, ning JavaScript.

Keerukamate projektide implementeerimiseks ning „ratta uuesti leiutamise“ vältimiseks, on turul pakkuda suur hulk teke, raamistikke ning platvorme. Antud vahendite kasutamine hõlbustab näiteks keerukamate andmestruktuuridega töötamist, projekti eraldihallatavateks mooduliteks jaotamist ning dünaamilist templiitimist. Käesoleva projekti raames otsustati arendamise hõlbustamiseks väiksemate teekide asemel kasutada eesrakenduse platvormi, kuna rakendus sisaldab suhtlust tagarakendusega, mittetriviaalset graafi kujul sektsiooni, dünaamilisi vaateid ning mittetriviaalset käitumist sisendi puhul.

Eesrakenduse platvormi valides jälgiti järgmisi omadusi:

- Lihtne paigaldus
- Komponendipõhine arhitektuur, skaleeruvus
- Töötamiskiirus
- Populaarsus

Loetletud kriteeriumeid ning mõningast varasemat kogemust arvesse võttes, sai eesrakenduse valitud JavaScript keelel põhinev raamistik Angular (töö kirjutamise hetkel versioon 7).

Angular raamistikul põhinevate rakenduste loomist hõlbustab raamistiku autorite poolt toetatud käsureatööriist Angular CLI. See toetab arendamisel järgmiseid ülesandeid:

- Arenduskeskkonna / esmase failistruktuuri ülesseadmine
- Moodulitega seotud operatsioonid, näiteks loomine
- Projekti ehitamine
- Rakenduse serveerimine lokaalselt

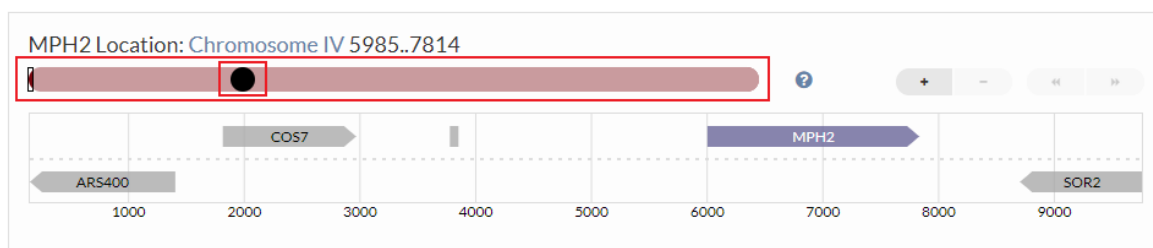
Antud projekti puhul oli suurimat kasu esmase ülesseadmise ning lokaalse serveerimise funktsionaalsustest [11].

Töö kirjutamise hetkel on Angular GitHub'i andmetel eesrakenduste JavaScript raamistike seas populaarsuselt kolmandal kohal pärast raamistikke Vue ning React [12].

Visualiseerimise teek

Andmebaasist päritud andmete visualiseerimiseks sai kaasatud eesrakendusse graafide kuvamiseks mõeldud JavaScript'il põhinev teek. Arvestades kohati ebastandardseid nõudeid, jäi otsingu tulemusena sõelale vaid üks teek, mis sobis oma standard-funktsionaalsuste poolest.

Nagu funktsionaalsetes nõuetes sai välja toodud, oli peamisteks nõueteks eraldi kihtide kuvamise võimalus horisontaalses tulpdiagrammis (joonis 2) ning et oleks võimalik kuvada eraldiseisvaid vahemikke, ilma et vahemikud peaksid olema ühendatud ühte pidevasse jadasse alates telje 0-koordinaadist.



Joonis 2. Geeni asukoha visualisatsioon SGD keskkonnas

Eelmainitud kriteeriumite põhjal sai valitud teek Highcharts JS [13]. Antud rakenduse jaoks kasutati teegi poolt pakutavat graafitüüpi „xseries“, mida on ka näha graafi initsialiseerimisel kasutatavast konfiguratsioonist (joonis 3).

```
7 Highcharts.chart(this.chartTarget.nativeElement, options: {
8   chart: {
9     type: 'xrange',
10    zoomType: 'x',
11  },
12  plotOptions: {
13    series: {
14      point: {
15        events: {
16          click: function (e) {
17            if (e.point.sgdId) {
18              window.open(e.point.sgdId, target: "_blank");
19            }
20          }
21        }
22      }
23    },
24    xrange: {
25      minPointLength: 1
26    }
27  },
28  tooltip: {
29    formatter: function() {
30      return AppComponent.formatTooltip(this.point);
31    }
32  },
33  yAxis: {
34    categories: chartCategories,
35    reversed: true,
36  },
37  series: [{
38    name: 'Chromosome length',
39    pointWidth: 22,
40    data: this.chartData,
41    dataLabels: {
42      enabled: false
43    },
44    type: 'xrange'
45  }],
46  legend: {
47    enabled: false,
48  }
49 });
```

Joonis 3. Lühendatud versioon graafi konfiguratsioonist

5.2 Tagarakendus

Veebirakenduste juures tagarakenduseks nimetatakse rakendust, mis töötab serveri peal ning võtab vastu erinevate klientide poolt saadetavaid näiteks HTTP päringuid. Sealjuures antud päringute peale tehakse päringuid tihti mõnda välisesse süsteemi, näiteks andmebaasi või teostatakse rakenduse siseselt ärioloogilisi andmete muundamisi. Siseste protsesside järel üldiselt tagastab tagarakendus kliendi esialgse päringu peale vastuse.

Sellised tagarakendused kirjeldavad üldiselt liidese, mida jälgides on võimalik sisukalt suhelda. Antud liides koosneb selgelt defineeritud lõpp-punktidest ning läbi nende kättesaadavatest ressurssidest.

Käesoleva projekti raames sai loodud tagarakendus vahendamaks päringuid andmebaasi suunas, mis hoiab antud rakenduse jaoks sobivale kujule teisendatud andmeid.

Nagu ka eesrakenduse puhul, on tagarakenduse realiseerimiseks olemas palju erinevaid raamistikke ning platvorme, mis toetavad arendamist. Valiku langetamisel jälgitakse kiirust, dokumenteeritust ning aktiivse kogukonna olemasolu.

A. Goeli artiklis [14] esitatakse töö kirjutamise hetkeks mõningad tuntud valikud:

- Django (Python) – kasutusel suurtes firmades nagu Google ning Youtube. Tähtsal kohal rakenduse turvalisus.
- Ruby on rails (Ruby) – Kasutusel firmades Airbnb ja GitHub. Rails on algajasõbralik ning teeb arendaja eest palju asju kulisside taga. Puuduseks on keerukas *live* keskkonna paigaldamine.
- Express.js (Node.js) – Trendikas platvorm tänu Node.js populaarsusele. Kasutusel samuti suurtes firmades, kuid puudu on laialtlevinud standardid.
- Laravel (PHP) – Suhteliselt noor võrreldes teiste väljatoodud raamistikega. Suure kogukonnaga, kuid jääb teistele valikutele kiiruse puhul alla.
- Spring (Java) – Aastast 2002 pärit raamistik. Põhineb tugevalt tüübitud Java'l. Mahuka dokumentatsiooni ning väga suure kogukonnaga.

Antud töö puhul sai valitud Java keelel ülesse ehitatud Spring raamistik, täpsemalt Spring raamistiku moodulid, mis on rohkem seotud veebirakendustega.

5.3 Andmebaas

Andmebaasi valimise protsessi suunasid implementeeritava rakenduse andmetega seotud toimingute eripärad:

- Kasutaja tegevuste peale andmeid vaid loetakse. Uute andmete kirjutamist andmebaasi ei toimu
- Andmed pärinevad välistest failidest ning uuenevad harva, seega lihtsaim viis andmete uuendamiseks oleks uute versioonidega allikatest baas uuesti ülesse ehitada

Kuna arendatavas rakenduses andmed ei muutu pärast esmast sisselugemist, siis võib öelda, et andmebaasi seisund ei ole kriitiline ühegi operatsiooni jaoks. Seega keskendutakse valikut tehes põhiliselt lugemisoperatsioonide kiirusele. Eelmainitu põhjal sai kaalutud mälusisese andmebaasi kasutamist ilma taastatavuseta.

Taastatavuse (*persistence*) all peetakse silmas võimekust rakenduse töökatkestuse puhul säilitada võimalikult hiline andmete seis andmebaasis.

Mälusiseste andmebaaside puhul on võimalik üles seada samatulemuslikku taastatavust võrreldes andmebaasidega, mis hoiavad oma andmeid stabiilsemates meediumites (näiteks kõvaketas). Selleks kasutatakse aeg-ajalt andmebaasi seisuga (*snapshot*) kirjutamist stabiilsele meediumile ning samuti hoitakse eraldi antud seisuga kaasaskäivat transaktsioonide logi (*transaction log*). Sellise lähenemise kasutamine on piisavalt kiire, et endiselt õigustada kiiruse kaalutlusel mälusisese andmebaasi kasutamist [15].

Antud rakenduse puhul taastatavust seadistada pole vaja, kuid võimalikke lisa-arendusi silmas pidades on aga hea, et seda funktsionaalsust on võimalik hiljem kasutusele võtta.

Valik sai langetatud *H2* andmebaasi kasuks, kuna seda on lihtne üles seada *Spring* raamistikku kasutava rakendusega ning ta täidab eelmainitud nõudeid. Samast kategooriast alternatiivideks sobiksid näiteks andmebaasid *Derby* või *HSQLDB* [16].

6. Andmete ettevalmistamine

Koostatav andmebaas sisaldab andmeid kolmest allikast. Siinkohal tuuakse välja mudel Java klasside näol, mis kirjeldab implementeeritava rakenduse andmebaasi struktuuri.

```
class GeneOntology {
    Long geneOntologyId;
    String name;
    List<Gene> genes;
}

class Gene {
    Long sgdId;
    String name;
    Integer chromosome;
    List<Location> locations;
}

class Location {
    Long id;
    Long start;
    Long end;
}
```

Edasi kirjeldatakse kasutatud andmeallikaid ning millised osad neist võeti kasutusse, et täita eelmainitud mudelit.

6.1 Geenide asukohad kromosoomidel

Geenide asukohad leiduvad SGD keskkonnas asuvas ressursis *orf_trans_all.fasta*.

FASTA formaati kasutatakse bioinformaatikas nukleotiidide ning valkude jadade kirjeldamiseks. Selle formaadi sisu koosneb jadade loetelust, kus esmalt esitatakse jada kirjeldus, millele järgneb kirjeldatav jada [17].

Antud ressursi puhul pole küll dokumenteeritud jadasid kirjeldavate ridade struktuur, kuid intuiitselt on märgata joonisel 4 välja toodud näite põhjal, et kirjetest on võimalik ekstraheerida järgmised omadused:

- *SGD ID* (geeni identifikaator SGD süsteemis)
- Geeni asukoht genoomil (kromosoomi number ning asukoha alg- ning lõpppunkt vastava kromosoomi aluspaaridel)

```
>YDR242W AMD2 SGDID:S000002650, Chr IV from 946807-948456, Genome Release 64-2-1,
Verified ORF, "Putative amidase"
MTVHSTWKEKVLKRDQLNSKIKDEWKLNSTTITRLKNDKKNLIKNNIDDLCSSENQITH
STIMALRQALEAKELSCHEITAAFCHRAALIHQVVNCLSEIMFSEALRLADYYDSNRPAI
LPPLYGIPISLKDQCNEGVDTSLGYLCRTFKPKTKNEESLIVSFLRDLGAIIFVKTTVP
SSMMATDTQSNTFGYTYNSINLSFSSGGSSGGEGSLIGAHGSLGLGTDIGGSIRIPSSY
QGLFGLKPTFGRVPYLRVDNSFEGRETIPSVIGPLARDLSDLRYFMSCVINICQPWVQDV
KCIPYHFDSSSTSKLHDNYVVGWYGDGVIDPPPSDIRALKTCEDLVNKTGKMAVKWEPS
SELSRELFDLANEADVADSGNEIKNEFEISGEPLLDILKPMVLENGRPPYTVNEWDDLTK
RVYNAQQLMRDYLSFPESERPDVVISPTTILMPFRPGDMLKTTLRYILLEFNVLFNPSLSI
PVGSVDCQIDGLMDTTSALNPEDKMIKTYWNDLIQSGEIDGFPIGLQVVSPTFNDNEVCK
FASWLFSKI*
```

Joonis 4. *orf_trans_all.fasta* failis esineva kirje näide

6.2 Geenide kuuluvus geeniontoloogia termini alla

Geeniontoloogia terminite ning nende alla kuuluvate geenide vaheline kuuluvus on kättesaadav ressursist *sgd.gaf* [18]. Geeniontoloogia konsortsium kirjeldab ka vastava formaadi definitsiooni [19], millest antud projekti jaoks pakuvad huvi formaadis sisalduvad tulbad:

- 2: *DB Object ID* (geeni *SGD ID*)
- 3: *DB Object Symbol* (geeni nimi)
- 5: *GO ID* (geeniontoloogia id)

6.3 Geeniontoloogiateg nimed

Seni loetletud allikad sisaldavad andmeid, mille puhul ontoloogiateg otsimiseks ning presenteerimiseks on sobinud vaid ontoloogia ID. Et kasutuskogemust rohkem inimsõbralikuks muuta, lisati andmetele juurde ka ontoloogiateg nimed.

Nimed on kättesaadaval ressursis *go_terms.tab* [20]. Antud faili puhul on defineeritud tulpadeks:

- *GOID*
- *GO_Term* (ontoloogiatermini nimi)
- *GO_Aspect* (millise ontoloogiateg on tegu: P (protsess), F (funktsioon), C (komponent))
- *GO_Term_Definition* (GO termini definitsioon)

Siinkohal on kaetud peatüki alguses väljatoodud mudeli täitmiseks vajaminevad andmeallikad. Edasi uuritakse järgneva etappeg ning võimalusi, kuidas arendatud rakendus lokaalsest arenduskeskkonnast viia seisu, kus rakendus oleks ligipääsetav avalikust veebist.

7. Rakenduse pakendamine, serveerimine ning majutus

Et arendatud rakendus töötaks mõne teenusepakkuja keskkonnas ning oleks avalikult kättesaadav veebis, tuleb tähelepanu pöörata ka rakenduse pakendamisele ning teenusepakkuja keskkonna ülesseadmisele.

Lihtsamate veebilehtede puhul võib see tähendada lihtsa HTTP serveri seadistamist keskkonnas ning staatiliste failide üleslaadimist. Tagarakendusega veebirakenduste puhul sõltuvalt kasutusel olevatest tehnoloogiatest tuleb keskkonda seadistada rohkem. HTTP serverile lisaks või selle asemel võib tulla ette näiteks veebikonteinerite või suuremate räämistikke paigaldamine.

Veebimajutuse teenusepakkujatel on eelmainitud protsessidest lähtuvalt palju erinevaid pakkumisi. Alustades ssh abil kaughallatavatest keskkondadest, kuhu on paigaldatud praktiliselt vaid operatsioonisüsteem ning lõpetades veebiteenustega, kus saab viisardit kasutades teha kogu vajaliku keskkonna konfigureerimise ning oma rakenduse üleslaadimise.

7.1 Veebi ning rakenduse serverid

Rakenduse serveriks peetakse tarkvara platvormi, mis hõlbustab tava- või ka veebirakenduste loomist ning pakub samas serveri keskkonda, kus rakendusi jooksutada. Masinas töötava rakenduse serveri peamine ülesanne on installeerida, hallata ning majutada endasse paigaldatud rakendusi, mis on arendatud vastava serveri jaoks.

Rakenduse arendaja seisukohast pakuvad rakenduse serverid olenevalt implementatsioonist erinevaid enamlevinuid liidestuste ning funktsionaalsusega seotud võimalusi lihtsasti tarbitaval kujul [21].

Java keskkonnas kirjeldab Eclipse Foundation rakenduse serveritele üldlevinuid standardeid Java EE spetsifikatsiooni näol. Erinevad tooted turul vastavad oma implementatsioonis antud spetsifikatsioonile kas osaliselt või täielikult. Siinkohal tasub aga märkida, et realsuses paljud Java EE spetsifikatsiooni osaliselt täitvad tooted suudavad täis-spetsifikatsioonile mittevastavusele vaatamata serveerida kompleksseid rakendusi ning tihtipeale saab spetsifikatsioonist puuduoleva toe lisada lihtsalt kolmanda osapoole tarkvaraga. Siinkohal toome välja V. Zukanovi artiklil [22] põhineva võrdluse neljast Java keskkonna rakenduse serveri implementatsioonist.

Tomcat

Apache Tomcat puhul on tegemist Java veebirakenduste puhul kirjutamise hetkel kõige populaarsema rakenduse serveriga. Tomcat'i puhul on tegemist ka näitega implementatsioonist, mis ei toeta täielikult Java EE spetsifikatsiooni. Korrektn oleks nimetada Tomcat'i veebiserveriks või servlet'ide konteineriks (servletid on üks osa Java EE spetsifikatsioonist). Osaline spetsifikatsioonile vastavus annab Tomcat'ile eelise paigalduse keerukuse juures ning tähendab ka väiksemat ruumivajadust installatsiooni jaoks. Populaarsusega käivad kaasas ka mahukas dokumentatsioon ning kogukond, kelle juurest võib lihtsalt abi saada.

Jetty

Eclipse Foundation'i poolt arendatav Jetty on samuti platform, mis ei toeta paljusid Java EE poolt kirjeldatud funktsionaalsusi. Tema eeliseks on see-eest väikene suurus, mis teeb Jetty'st hea valiku, kui süsteem, kuhu seda paigaldatakse, on väikese mälumahuga. Samuti sobib ta teiste toodete sisse pakendamiseks.

GlassFish

Antud puhul on tegemist Oracle poolt hallatava igat Java EE spetsifikatsiooni poolt kirjeldatud funktsionaalsust toetava platvormiga. Kuna Oracle ning Java EE on omavahel tihedalt seotud, kasutatakse tihti GlassFish'i justnimelt uute Java EE featuuride reklaamimiseks kolmandatele osapooltele.

WildFly

Varasemalt tuntud JBoss Application Server nime all, toetab Red Hat'i poolt arendatud WildFly sarnaselt GlassFish'iga tervet Java EE spetsifikatsiooni. Käesoleva platvormi eelis on Red Hat'i poolt pakutavas lihtsas migratsioonis sama tootja kommertstoega alternatiivsele pakkumisele, nimelt JBoss Enterprise Application Platform'i jaoks.

Valiku langetamisel tuleb arvesse võtta konkreetse projekti vajadusi. Keerukamate platvormide puhul tuleb üldiselt kaasa ka platvormi enda administreerimisega lisakompleksust.

Tasub panna tähele ka asjaolu, et korrektselt disainitud tarkvaralahenduse puhul peaks implementatsioon sõltuma minimaalselt platvormidest, mida kasutatakse rakenduse majutamisel või erinevate süsteemide liitepunktides. Seetõttu ei tasu ilma eelneva kaalutluseta

võtta esialgu kasutusele näiteks GlassFish raamistikku. Vajadusel peaks saama tulevikus muudatuse teha ilma suurema vaevata.

Käesoleva projekti puhul sai enamus konteinerite, toetavate raamistike ning pakendamisega seotud nüansid lahendatud Spring raamistikuga tugevalt ühte takti areneva projektiga Spring Boot. Antud projekti abiga sai suurema mureta pakendada rakenduse koos veebiserveriga ühte käivitatavasse konteinerisse. Siinkohal toome välja osa Spring Boot projekti veebilehel [23] välja toodud põhilistest pakutavatest featuuridest:

- Võimalus luua eraldiseisvaid Spring raamistikul põhinevaid rakendusi
- Võimalus manuserverina kaasa pakkida Tomcat, Jetty või Undertow
- Võimalus kasutada Spring raamistiku ning kolmanda osapoole liidestuste jaoks automaatset konfiguratsiooni loomist
- Võimalus vältida XML konfiguratsiooni loomist, mis käib tavaliselt kaasas konteinerite ja rakenduse serverite jaoks oma rakenduse ettevalmistamisega

Kasutades Spring Boot projekti põhifunktsionaalsusi, saab lihtsa kompileerimiskäsu tulemuseks käivitatava Java arhiivifaili. Antud arhiivifaili käivitamise tulemusena serveeritakse arendatud rakendust läbi Tomcat manuserveri. Edasi antakse ülevaade mõnest valikust toodete kohta, kus majutada arendatud rakendust ning puudutatakse lühidalt ka valitud keskkonna konfigureerimist.

7.2 Majutus

Rakendusele majutusvalikuid otsides tuleb lähtuda arendatava rakenduse struktuurist või pakendusstrateegiast. Antud projekti puhul lähtutakse manuserverit sisaldava käivitatava Java arhiivifaili (edaspidi jar) kasutamise seisukohast. Kuna selline lahendus on üpris eraldi-seisev ning platvormidest sõltumatu, siis valikut langetades eelistatakse nõ „infrastruktuur teenusena“ teenuseid „platvorm teenusena“ teenuste asemel. Siinkohal tutvustatakse kahte Amazon AWS keskkonna poolt pakutavat teenust, millega on võimalik hõlpsalt sellist rakendust majutada.

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 [24] puhul on tegemist teenusega, mille abil on võimalik kergelt erinevate omadustega arvutusvõimekust tellida. Arendaja või administraatori jaoks tähendaks see üldjuhul läbi EC2'e erinevate serverikeskkondade tellimist, ülesseadmist ning olemasolevate

instantside manageerimist. Teenuse eest tasumine toimub vastavalt kasutatud ressursi tüübist ning mahust. Antud variant kaldub rohkem „infrastruktuur teenusena“ kategooriasse.

AWS Elastic Beanstalk

Elastic Beanstalk [25] pakub ühe lisa abstraktsuskihi tavaliste EC2 instantside juurde. Nimelt peale arvutusressursi ning operatsioonisüsteemi pakutakse siinkohal ka võimalust *out-of-the-box* seadistada erinevat serveritarkvara (Apache HTTP serverist GlassFish rakenduse serverini) ja liidestusi näiteks andmebaasiteenustega. Lisaks pakutakse automaatset arendaja rakenduse serverisse paigaldamist, koormuse tasakaalustamist jpm. Elastic Beanstalk poolt pakutavad lisateenused ise eraldi ei ole tasulised, küll aga võivad nad enda all paiknevaid EC2 ressursse teatud seadistuste puhul rohkem tarbida, millest vastavalt muutuks ka teenuse kasutamise maksumus.

Käesoleva rakenduse jaoks sai majutus realiseeritud ühe EC2 instantsi näol. Siinkohal tuuakse välja lühike ülevaade etappidest, mis sai tehtud, et jõuda avalikult ligipääsetava jooksva rakenduseeni:

- EC2 instantsi (t2.micro tüüpi) tellimine Ubuntu 18.04 operatsioonisüsteemiga ning ssh ligipääsuga (Amazoni enda nõ tulemüürireegli abil ning avaliku võtme esitamise läbi)
- Rakenduse arhiivifaili serverisse ülesse laadimine läbi ssh
- Serveri tarkvarapakettide uuendamine
- Serverisse OpenJDK JDK 8 installeerimine
- Serverisse iptables-persistent installeerimine
- Serveris pordisuunamise konfigureerimine iptables käsu abil rakenduse ning serveri pordi 80 vahel
- Pordisuunamiste salvestamine iptables-save abil
- Rakenduse jar faili käivitamine taustaprotsessina
- Läbi EC2 manageerimisvaate tulemüürireegli lisamine, lubades liikluse instantsi port 80 peal

7.3 Kasutajaliidese kirjeldus

Loodud kasutajaliides koosneb neljast plokist: päisest koos sissejuhatava tekstiga (joonis 5), sisendlahtrist (joonis 5), otsingutulemuse graafilisest väljundist (joonis 5) ning tabelkujul väljundist (joonis 6).



Joonis 5. Rakenduse kasutajaliides lehe algusest kuni graafini

| Result table for <u>histone acetyltransferase complex</u> (GO:0000123) | | | | | Copy table to clipboard as CSV |
|--|-------|------------|--------|--------|--------------------------------|
| Gene | SGDID | Chromosome | Start | End | |
| SAS3 | 148 | 2 | 119379 | 121874 | |
| HAT2 | 782 | 5 | 47168 | 48373 | |
| TRA1 | 1141 | 8 | 302761 | 313995 | |
| EAF6 | 3842 | 10 | 581914 | 582255 | |
| HIF1 | 3945 | 12 | 99044 | 100201 | |
| SAS2 | 4734 | 13 | 522329 | 523345 | |
| ESA1 | 5770 | 15 | 792531 | 793868 | |
| HAT1 | 5922 | 16 | 554605 | 555729 | |
| EAF3 | 6227 | 16 | 608826 | 610031 | |

<https://www.yeastgenome.org/locus/S000000782>

Joonis 6. Otsingutulemuse tabelkujul väljund

Siinkohal võetakse ka kokku tehnoloogiate valikute, nende juures otsuste langetamise ning erinevate implementatsioonidetailidega seotud ülevaade. Edasi uuritakse rakenduse testimist.

8. Testimine

Tulenevalt rakenduse nõuetest, esitatakse siinkohal kahed kolmanda osapoole tööriista abil läbiviidud analüüsi tulemused. Käsitletakse esmalt veebibrauserite tuge, seejärel jõudlust ning lõpuks tagasisidet kasutajatelt..

8.1 Veebibrauserite tugi

Rakenduse nõuded määrasid, et rakendus peab ühilduma enimkasutatud lauaarvutite ning mobiilsete brauseritega. Joonisel 7 esitatud testi tulemustest on näha, et uuemate ning populaarsemate brauseritega probleeme ei ole. Küll aga esineb probleeme lehe laadimisel vanemate brauserite IE7, IE8, IE9 ning Firefox 3.6 kasutamisel.



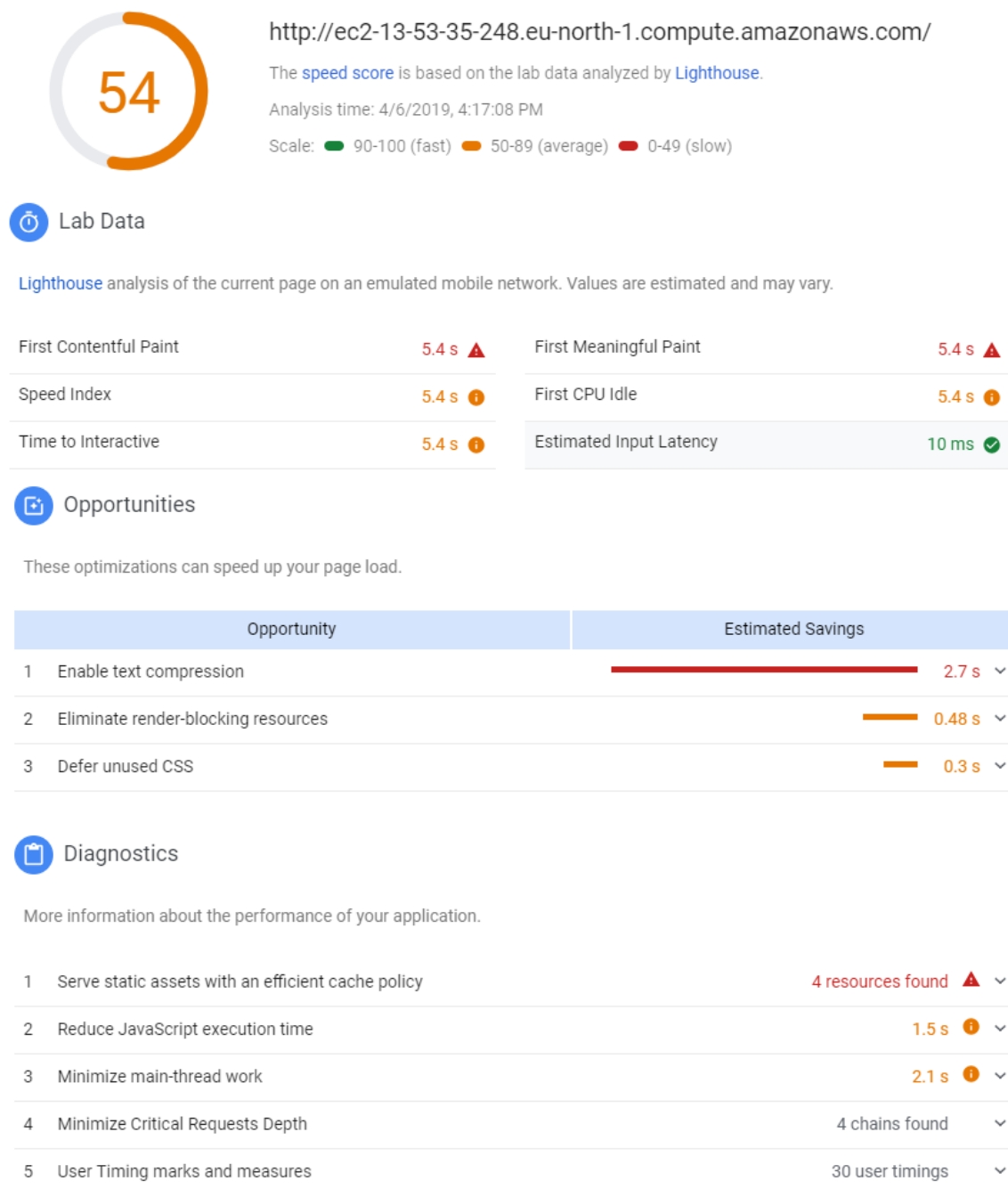
Joonis 7. Brauseritoe testitulemused

Veebibrauserite toe testimiseks kasutati tarkvara BrowserStack [26].

8.2 Jõudlus

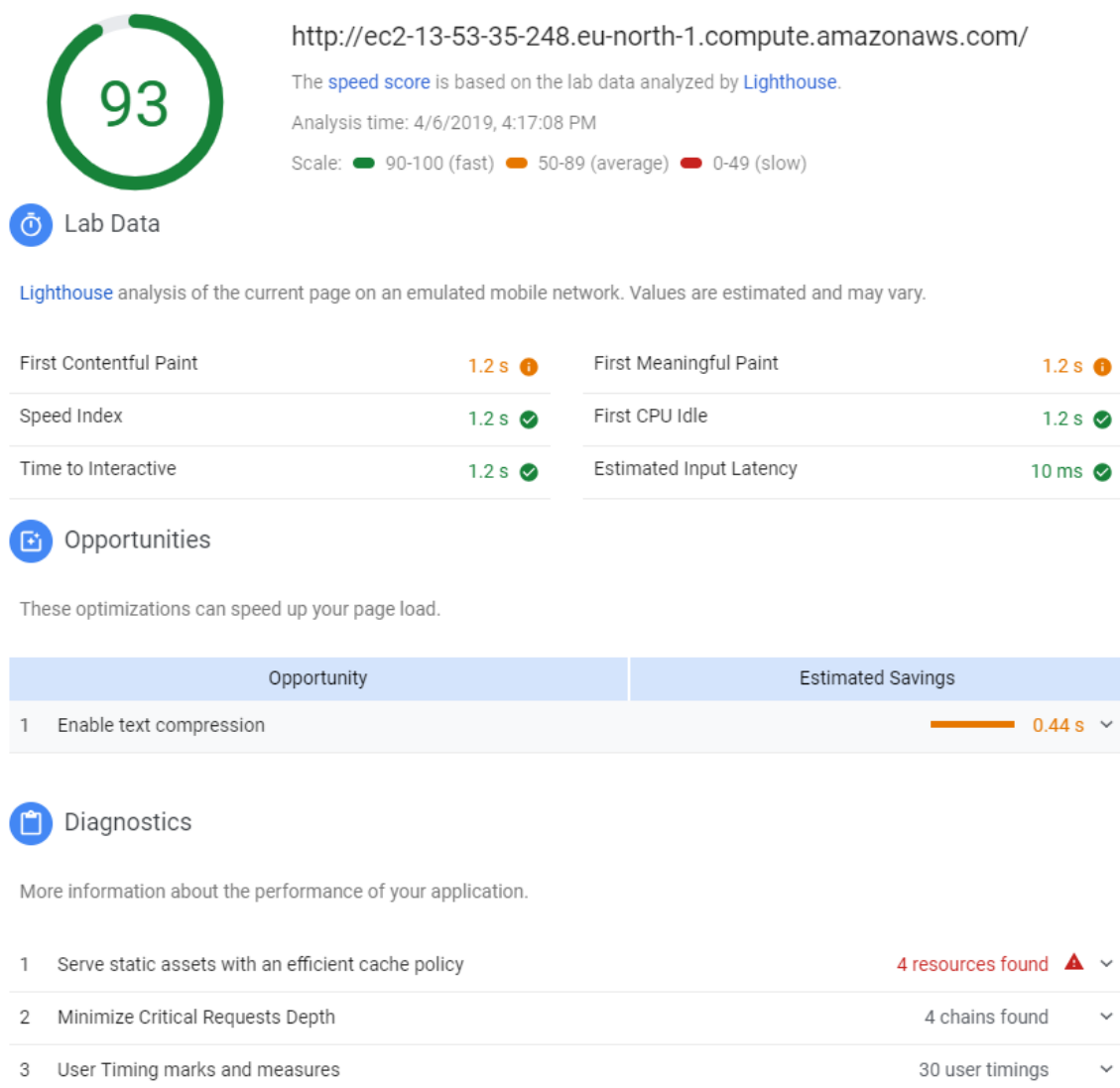
Tulenevalt jõudlusega seotud nõuetest, viidi läbi rakenduse peal analüüs Google'i tootega PageSpeed Insights [27]. Lisaks analüüsi tulemustele, pakub antud tööriist ka soovitusi jõudluse tõstmiseks. Analüüsi tulemustes esitatakse kaks perspektiivi: joonisel 8 mobiilid ning joonisel 9 lauaarvutid.

Mobiilide kategooria



Joonis 8. Jõudlustesti tulemused mobiilide kategoorias

Lauaarvutite kategooria



Joonis 9. Jõudlustesti tulemused lauaarvutite kategoorias

Kuna enamus kasutussituatsioone hõlmavad eeldatavasti veebilehe kasutamist lauaarvutite kategooriasse langevate seadmetega, siis saab analüüsi tulemuse lugeda rahuldavaks.

8.3 Tagasiside kasutajatelt

Kahelt kasutajalt saadud tagasiside oli üldiselt positiivne ning leiti, et tööriist täidab oma eesmärgi, kiirendades töö poolt esitatava probleemi poolt kirjeldatavat protsessi. Lisaks toodi välja aspekte ning funktsionaalsusi, mida võiks rakendus tulevikus toetada. Nendeks on:

- Võimalus pärida ka üksikuid valke / geene nende nime järgi (näiteks „TRA1“)
- Võimalus laadida tulemuse tabel alla failina populaarsemates formaatides (XLSX, CSV jne)
- Võimalus pärida andmeid ka KEGG'st (Kyoto entsüklopeedia geenidest ning genoomidest [28]), kus paikneb sarnasel kujul annotatsioonid.

Siinkohal võetakse kokku testimise peatükk. Võib öelda, et testid olid edukad, kuna rakendus toetab enimkasutatuid veebibrausereid, rakenduse jõudlustulemused on head ning kasutajad leidsid, et tööriist tuleb kasuks. Positiivne on ka asjaolu, et kasutajad leidsid funktsionaalsusi, mida rakendus võiks tulevikus toetada. Autori arvates näitab see initsiatiivi ning huvi tööriista käekäigu suhtes tulevikus.

9. Kokkuvõte

Käesoleva töö eesmärgiks oli tuvastada pärmi genoomis asuvate geeniontoloogia terminite ning nende alla kuuluvate geenide asukohtade vaheliste kaardistuste kohta andmeallikad ning nende põhjal koostada andmebaas koos veebisaidiga. Töö käigus anti esmalt ülevaade probleemi olemusest pärmi genoomiga töötavate inimeste seisukohast ning olemasolevate tööriistade limitatsioonidest. Seejärel tutvustati erinevaid tehnoloogiate ning teenuste valikuvariante uue tööriista implementeerimiseks. Iga valiku juures esitati ka implementeerimisel kasutusele võetud variant. Lõpetuseks anti ülevaade loodud rakenduse majutamistest ning püstitatud nõuete vastavusele testimisest.

Käesoleva bakalaureusetöö raames valmis veebirakendus, kus implementeeriti eraldi tagarakendus andmebaasiga ning eesrakendus. Tagarakendus põhines Spring Boot platvormil, mis baseerub Spring raamistikul, mis omakorda põhineb Java keelel. Andmebaasi puhul langetati otsus mälu sisese andmebaasi H2 kasuks. Eesrakenduse jaoks võeti kasutusele JavaScript keelel põhinev platvorm Angular. Majutuse jaoks valiti Amazon AWS teenuste seast serverikeskkond EC2 toote instantsi näol, kuhu paigaldatud operatsioonisüsteemiks valiti Ubuntu 18.04.

Käesoleva bakalaureusetöö edasiarendusena oleks võimalik lisada erinevaid kasutatavust parendavat funktsionaalsust, näiteks otsingutulemuste eksportimise võimalust standardformaattidesse ning muud sarnast. Lisaks on avaldanud juhendaja soovi laiendada toetatud andmeallikaid. Nimelt geeniontoloogia terminitele lisaks, võiks rakendus toetada ka KEGG'is (Kyoto entsüklopeedia geenidest ning genoomidest) sisalduvaid sarnaseid kaardistusi.

10. Viidatud kirjandus

- [1] Eukaryote. *Encyclopaedia Britannica*.
<https://www.britannica.com/science/eukaryote> (30.03.2019)
- [2] The Gene Ontology Consortium Gene Ontology: tool for the unification of biology. *Nature Genetics*, 2000.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3037419/pdf/nihms-269796.pdf>
(30.03.2019)
- [3] About the GO. *Gene Ontology Consortium*.
<http://geneontology.org/docs/introduction-to-go-resource/> (30.03.2019)
- [4] Introduction to GO annotations. *Gene Ontology Consortium*.
<http://www.geneontology.org/docs/go-annotations> (29.04.2019)
- [5] Engel, S.R. et al. The reference genome sequence of *Saccharomyces cerevisiae*: then and now. *G3 (Bethesda)*, 2013.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3962479/> (04.05.2019)
- [6] *Saccharomyces* genome database. <http://www.yeastgenome.org/> (02.05.2019)
- [7] Skrzypek M.S., Hirschman J. Using the *Saccharomyces* Genome Database (SGD) for analysis of genomic information. *Curr Protoc Bioinformatics*, 2011.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3191907/pdf/nihms323315.pdf>
(02.05.2019)
- [8] YeastMine. *Saccharomyces* genome database.
<https://yeastmine.yeastgenome.org/yeastmine/begin.do> (02.05.2019)
- [9] Jbrowse. *Saccharomyces* genome database. <https://browse.yeastgenome.org>
(02.05.2019)
- [10] Gene Ontology Term Finder. *Saccharomyces* genome database.
<https://www.yeastgenome.org/cgi-bin/GO/goTermFinder.pl> (02.05.2019)
- [11] Getting started. *Angular*. <https://angular.io/guide/quickstart> (02.05.2019)
- [12] Front-end JavaScript frameworks. *GitHub*. <https://github.com/collections/front-end-javascript-frameworks> (02.05.2019)
- [13] Highcharts. <https://www.highcharts.com/> (02.05.2019)

- [14] Goel A. Top 10 Web Development Frameworks in 2019. *Hackr.io Blog*, 2019.
<https://hackr.io/blog/top-10-web-development-frameworks-in-2019> (02.05.2019)
- [15] Anikin D. What an in-memory database is and how it persists data efficiently.
Medium, 2016. <https://medium.com/@denisanikin/what-an-in-memory-database-is-and-how-it-persists-data-efficiently-f43868cff4c1> (02.05.2019)
- [16] H2 Database Engine. <http://www.h2database.com/html/main.html> (02.05.2019)
- [17] What is FASTA format? *Zhang Lab, University of Michigan*.
<https://zhanglab.ccmb.med.umich.edu/FASTA/> (02.05.2019)
- [18] GO annotation files. *Gene Ontology Consortium*.
<http://current.geneontology.org/annotations/> (02.05.2019)
- [19] GO Annotation File (GAF) format 2.1. *Gene Ontology Consortium*.
<http://geneontology.org/docs/go-annotation-file-gaf-format-2.1/> (02.05.2019)
- [20] Files with information curated or otherwise assembled by the SGD staff.
Saccharomyces genome database.
<https://downloads.yeastgenome.org/curation/literature/> (02.05.2019)
- [21] Hellard B., Moore-Colyer R. What is an application server? *ITPRO*, 2018.
<https://www.itpro.co.uk/strategy/29643/what-is-an-application-server> (02.05.2019)
- [22] Zukanov V. Top Java Application Servers: Tomcat vs. Jetty vs. GlassFish vs. WildFly. *Stackify*, 2018. <https://stackify.com/tomcat-vs-jetty-vs-glassfish-vs-wildfly/> (02.05.2019)
- [23] Spring Boot. <https://spring.io/projects/spring-boot> (02.05.2019)
- [24] Amazon EC2. <https://aws.amazon.com/ec2/> (02.05.2019)
- [25] AWS Elastic Beanstalk. <https://aws.amazon.com/elasticbeanstalk/> (02.05.2019)
- [26] Usage of web servers. *W3Techs*.
https://w3techs.com/technologies/overview/web_server/all (02.05.2019)
- [27] PageSpeed Insights. *Google Developers*.
<https://developers.google.com/speed/pagespeed/insights/> (02.05.2019)
- [28] KEGG: Kyoto Encyclopedia of Genes and Genomes. <https://www.genome.jp/kegg/> (05.05.2019)

Lisad

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Martin Põhjakivi,

(autori nimi)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Pärmi genoomi funktsioonide visualiseerimine veebirakenduses,

(lõputöö pealkiri)

mille juhendajad on Rahul Kumar ning Sven Aller,

(juhendajate nimed)

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Martin Põhjakivi

10.05.2019